

## Research Article

# Low-Cost SCADA System Using Arduino and Reliance SCADA for a Stand-Alone Photovoltaic System

Ibrahim Allafi  and Tariq Iqbal 

*Department of Electrical Engineering, Faculty of Engineering, Memorial University, St. John's, NL, Canada*

Correspondence should be addressed to Ibrahim Allafi; [ia1487@mun.ca](mailto:ia1487@mun.ca)

Received 2 April 2017; Accepted 27 July 2017; Published 2 May 2018

Academic Editor: Sundaram Senthilarasu

Copyright © 2018 Ibrahim Allafi and Tariq Iqbal. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

SCADA (supervisory control and data acquisition) systems are currently employed in many applications, such as home automation, greenhouse automation, and hybrid power systems. Commercial SCADA systems are costly to set up and maintain; therefore those are not used for small renewable energy systems. This paper demonstrates applying Reliance SCADA and Arduino Uno on a small photovoltaic (PV) power system to monitor the PV current, voltage, and battery, as well as efficiency. The designed system uses low-cost sensors, an Arduino Uno microcontroller, and free Reliance SCADA software. The Arduino Uno microcontroller collects data from sensors and communicates with a computer through a USB cable. Uno has been programmed to transmit data to Reliance SCADA on PC. In addition, Modbus library has been uploaded on Arduino to allow communication between the Arduino and our SCADA system by using MODBUS RTU protocol. The results of the experiments demonstrate that SCADA works in real time and can be effectively used in monitoring a solar energy system.

## 1. Introduction

For several hundred years, fossil fuels have been consumed as the main source of energy on Earth. As a result, they are now experiencing rapid depletion. Researchers and scientists who understand the importance of renewable energy have dedicated their efforts to the research, expansion, and deployment of new energy sources to replace fossil fuels.

Photovoltaics (PV) are an important renewable energy sources. Also called solar cells, PV are electronic devices that can convert sunlight directly into electricity. The modern forms of PV were developed at Bell Telephone Laboratories in 1954 [1]. Despite their promising performance, PV have some limitations, such as depending on factors like longitude, latitude, and weather and being limited to daytime hours to generate power [2].

The SCADA system is software that has been installed in several sites to monitor and control processes, and it is called telemetry importance [3, 4]. SCADA can monitor real-time electrical data measurements of solar module and batteries and collect data from wind turbines, such as the condition of the gearbox, blades, and electric system [5, 6]. Moreover,

the sun-tracker system has also used the SCADA system to observe the solar insolation and movement of the sun [6].

These days, commercial companies are widespread for monitoring systems such as photovoltaic systems. However, those are quite expensive. For example, SMA Company is a German Company, and it was founded in 1981. It has many products. Some of them are related to monitoring and controlling, for example, Sunny View. It can show all of your system data in good condition, and we can read all data clearly. However, the major problem is that the device is costly; it costs about CA \$793 [7, 8].

A previous study also shows a data acquisition and visualization system, with storage in the cloud, and it has been applied on a photovoltaic system. In addition, this design was based on embedded computer, and it connected with PV inverters by using RS485 standard, and microcontroller is to read climate sensors but sensors have used web system to show data [9].

Also, a study shows a low-cost monitoring system, it is presented in [10]. The system has determined losses in energy production. The paper is based on multiple wireless sensors and low cost, and it used voltage, current, irradiation, and



FIGURE 1: Solar panels on the roof of engineering building.

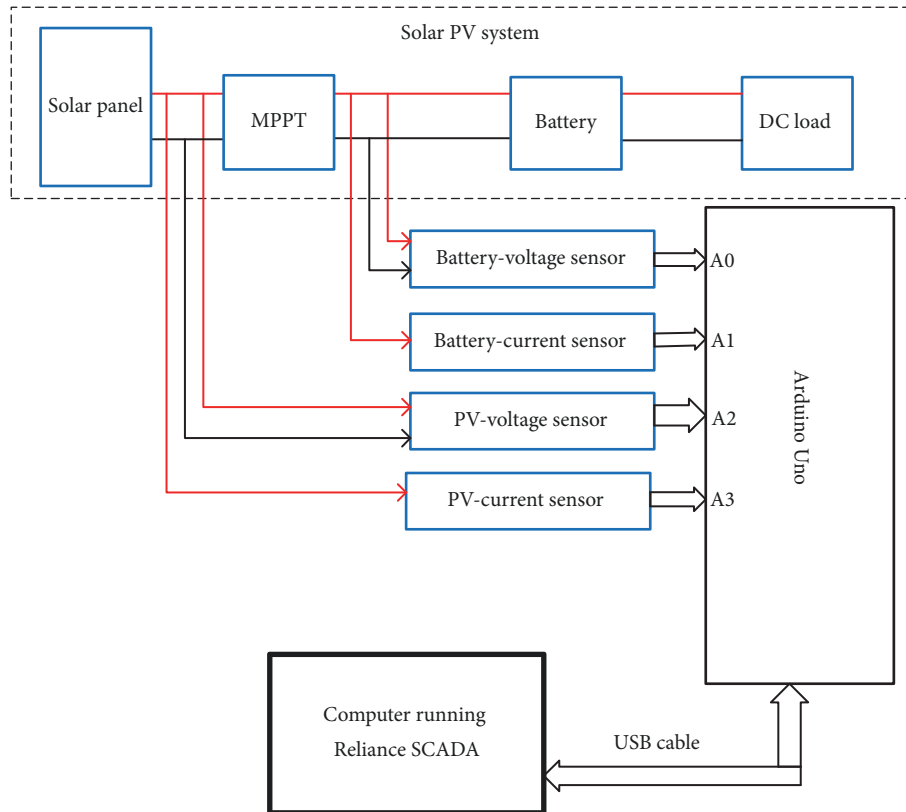


FIGURE 2: Hardware configuration of SCADA system.

temperature sensors which are installed on PV modules as well.

In this paper, designed SCADA system is of lower price compared with commercial SCADA system, and it delivers the same performance. In order to test this work, the SCADA system is employed for monitoring the parameters of solar energy systems (photovoltaic) in real time, which consist of a solar module, MPPT, and batteries. The parameters are the current and voltage of the photovoltaic (PV) system and the current and voltage of the battery. Data acquisition system is by Arduino controller and sensors. All data are sent to a PC and are shown on a user interface designed by Reliance SCADA. The data are saved on a computer as an Excel file as well. This allows users and operators to monitor the parameters of the PV system in real time. The components of the SCADA system in this paper consist of two parts: hardware and software.

## 2. Hardware Design

The proposed Reliance SCADA is designed to monitor the parameters of a small PV system. It is installed at the Department of Electrical Engineering, Memorial University, St. John's, Canada. Figure 1 shows 12 solar panels up to 130 watts and 7.6 amp. Two solar modules are connected in parallel. Therefore, the system shown in Figure 1 consists of 6 sets of 260 watts each.

The Reliance SCADA system was designed to be of low cost and can be expanded or modified without the need for major hardware changes in the future. Basic elements of the design are an Arduino Uno controller and sensors, as shown in Figure 2.

**2.1. Arduino Uno Microcontroller.** Arduino Uno is open-source hardware that is relatively easy to use. Figure 3 shows

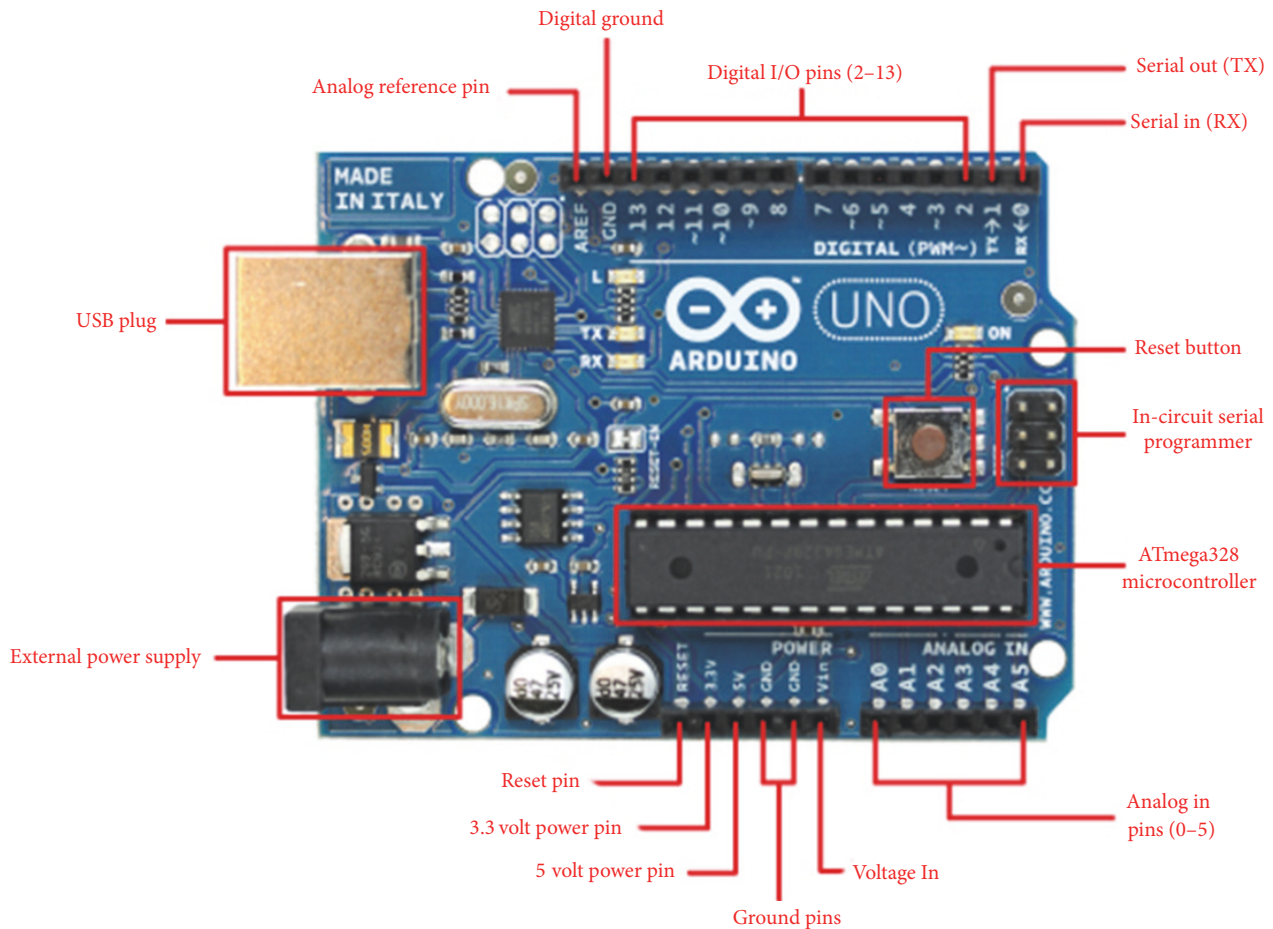


FIGURE 3: Arduino Uno R3.

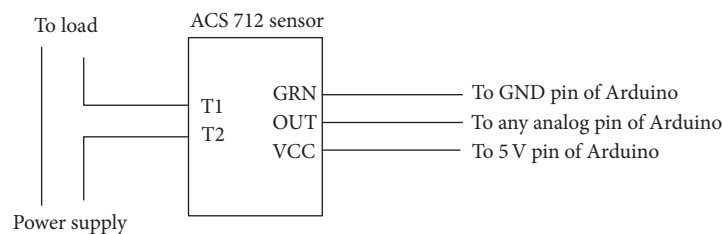


FIGURE 4: Connection drawing of current sensor.

Arduino Uno, while Table 1 shows specifications for the hardware. The license gives permission to anyone to improve, build, or expand Arduino. The original Arduino and its enhancement environment were founded in 2005 in Italy at the Smart Project Company. It has 14 digital input/output pins, 6 of which can be used as analog input/output [11].

**2.2. Current Sensor.** Current sensors for DC currents must be able to measure a range of currents for PV and batteries between 0 A and 20 A. In this work, an ACS 712 sensor is used for sensing the current. It is designed to be easily used with any microcontroller, such as Arduino. The sensors are based on the Allegro AC712ELC chip. The scale value of ACS 712, which is used in this design, is 20 amp, which is appropriate

for sensing current. Two sensors are installed: one is installed before MPPT to measure the PV current and the other is installed after MPPT to measure the battery current. Figure 4 demonstrates how it connects in an electrical circuit with Arduino Uno.

**2.3. Voltage Sensor.** The function of voltage sensors, which are based on just two resistors, is to turn a large voltage into a small one. In this work, the voltage sensor is a 25 V-sensor with two resistors of 30 K $\Omega$  and 7.3 K $\Omega$ . The maximum voltage of either PV or battery is 25 V, so this sensor is appropriate. The output of the voltage sensor is between 0 V and 5 V. This scale is suitable to the Arduino analog inputs. In this experiment, we need two voltage sensors: one is installed

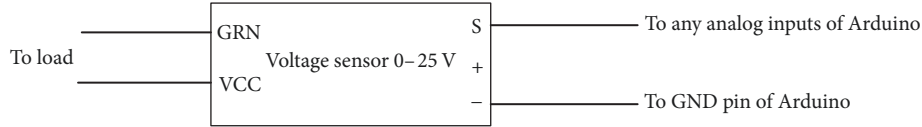


FIGURE 5: Connection drawing of current sensor.

TABLE 1: Specifications of Arduino board.

Specifications	
Microcontroller	ATmega328
Operating voltage	5 V
Input voltage	7–12 V
Input voltage (limits)	6–20 V
Digital I/O pins	14 (6 PWM outputs)
Analog input pins	6
DC current per I/O pin	40 mA
DC current for 3.3 V pin	50 mA
Flash memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock speed	16 MHz

before MPPT to measure the PV voltage and the other is installed after MPPT to measure the battery voltage. Figure 5 demonstrates how it connects in an electrical circuit with Arduino Uno.

### 3. Hardware Setup

Figure 6 shows the hardware setup designed for the SCADA system.

### 4. Software Requirements

**4.1. Arduino IDE.** IDE is open-source software that features easy-to-write code that can be uploaded to any board. In this work, we needed to upload a new library on IDE to make a configuration between Arduino Uno and SCADA software by MODBUS RTU protocol. Figure 7 shows how the system works and also shows the code that has been burned on Arduino Uno.

(A) *Flow Chart.* See Figure 7.

(B) *Code.* The code has some main functions such as `setup()` (it is called once when the sketch starts) and `loop()` (it is called over and over and is heart of sketch). The most important in the code are libraries mentioned initially: `regBank.setId()` command, `regBank.add()`, and `regBank.set()`. The purpose of libraries is to connect between Arduino Uno and Reliance SCADA software by MODBUS RTU protocol. `regBan.setId()` is used to define MODBUS to work as slave. `regBank.add()` command is used to define addresses of registers which are used to send data to Reliance SCADA on computer. In this work, the addresses were from 30001 to 30005 as mentioned



FIGURE 6: Hardware setup of SCADA system.

in the code, while `reBank.set()` command is used to write data on the previous addresses.

```
#include <modbus.h>
#include <modbusDevice.h>
#include <modbusRegBank.h>
#include <modbusSlave.h>

modbusDevice regBank;
modbusSlave slave;

float out1 = 0.0; float out2 = 0.0; float out3 = 0.0; float
out4 = 0.0; float AIO; float AI1; float AI2; int AI3; float
vpv = 0.0; float vbatt = 0.0; float eff = 0.0; float p1 =
0.0; float p2 = 0.0; int mVperAmp = 100; int ACSoffset
= 2500; double Voltage2 = 0; double Voltage3 = 0;
double Ampspv = 0; double Ampsbatt = 0; float R1 =
30000.0; float R2 = 7500.0; void setup(){
Serial.begin(9600);
regBank.setId(10); ///Set Slave ID
regBank.add(30001); regBank.add(30002); regBank
.add(30003); regBank.ad
d(3 0004); regBank.add(30005);
slave._device = &regBank;
slave.setBaud(9600);
pinMode(0,INPUT); pinMode(1,INPUT); pinMode(2,
INPUT); pinMode(3,I
```

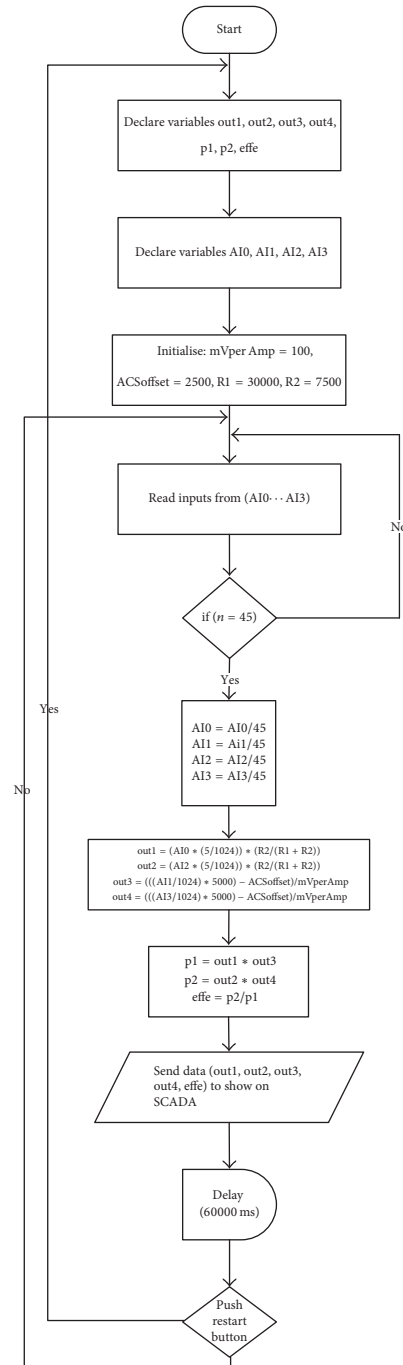


FIGURE 7

```

NP UT); }
void loop(){
while(1){
int AIO = analogRead(0); int AI1 = analogRead(1); int
AI2 = analogRead(2); int AI3 = analogRead(3); vpv =
(AIO * 5.0)/1024.0;
out1 = vpv/(R2/(R1+R2));
vbatt = (AI2 * 5.0)/1024.0;

```

```

out2 = vbatt/(R2/(R1+R2));
Voltage2 = (AI1/1024.0) * 5000;
Ampspv = ((Voltage2 - ACSoffset)/mVperAmp); Voltage3 = (AI3/1024.0) * 5000;
Ampsbatt = ((Voltage3 - ACSoffset)/mVperAmp); p1 = vpv * Ampspv;
p2 = vbatt * Ampsbatt;
eff = p2/p1;

```



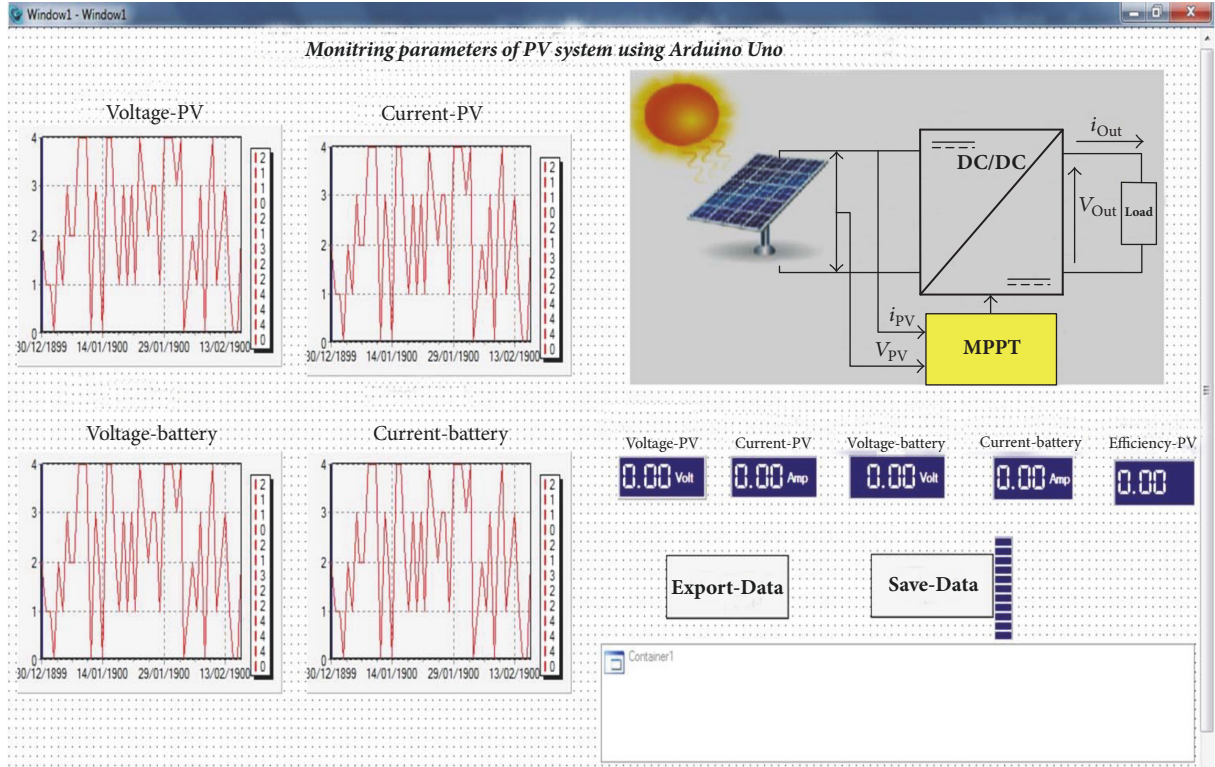


FIGURE 8: User interface of SCADA system.

```

Ampspv=abs(Ampspv);
Ampsbatt=abs(Ampsbatt);
out1=out1*100;
Ampspv= Ampspv*(100);
out2=out2*100;
Ampsbatt=Ampsbatt*(100);
eff=eff*100;
regBank.set(30001, (word) out1);
regBank.set(30002, (word) Ampspv);
regBank.set(30003, (word) out2);
regBank.set(30004, (word) Ampsbatt);
regBank.set(30005, (word) eff);
delay(60000)
slave.run();}}

```

**4.2. Reliance SCADA.** Reliance software is employed in numerous technologies for monitoring and controlling systems. It can also be used for connecting to a smartphone or the web. Reliance is used in many colleges and universities around the world for education or scientific research purposes [12]. Figure 8 shows a user interface designed by Reliance SCADA software to monitor the parameters of the photovoltaic system.

The user interface has four real-time trends and four display icons to show values as digital numbers. In addition, it

TABLE 2: Allocation of MODBUS address for MODBUS RTU.

Number	Variable name	MODBUS RTU address
(1)	Voltage of photovoltaic system	0
(2)	Current of photovoltaic system	1
(3)	Voltage of battery	2
(4)	Current of battery	3
(5)	Efficiency of MPPT	4

has two buttons and a container. These features are discussed in Results and Discussion.

## 5. Communication System

MODBUS library is added to Arduino Uno to allow communication with Reliance SCADA via a USB cable using MODBUS RTU protocol. Table 2 shows the allocation of MODBUS address for MODBUS RTU on Reliance SCADA software, with the MODBUS address for Arduino Uno mentioned in the Arduino code.

## 6. Cost of the SCADA System

Most factories that use several systems are looking for a low cost SCADA system to monitor and control their systems remotely. In this paper, the components used are quiet cheap.

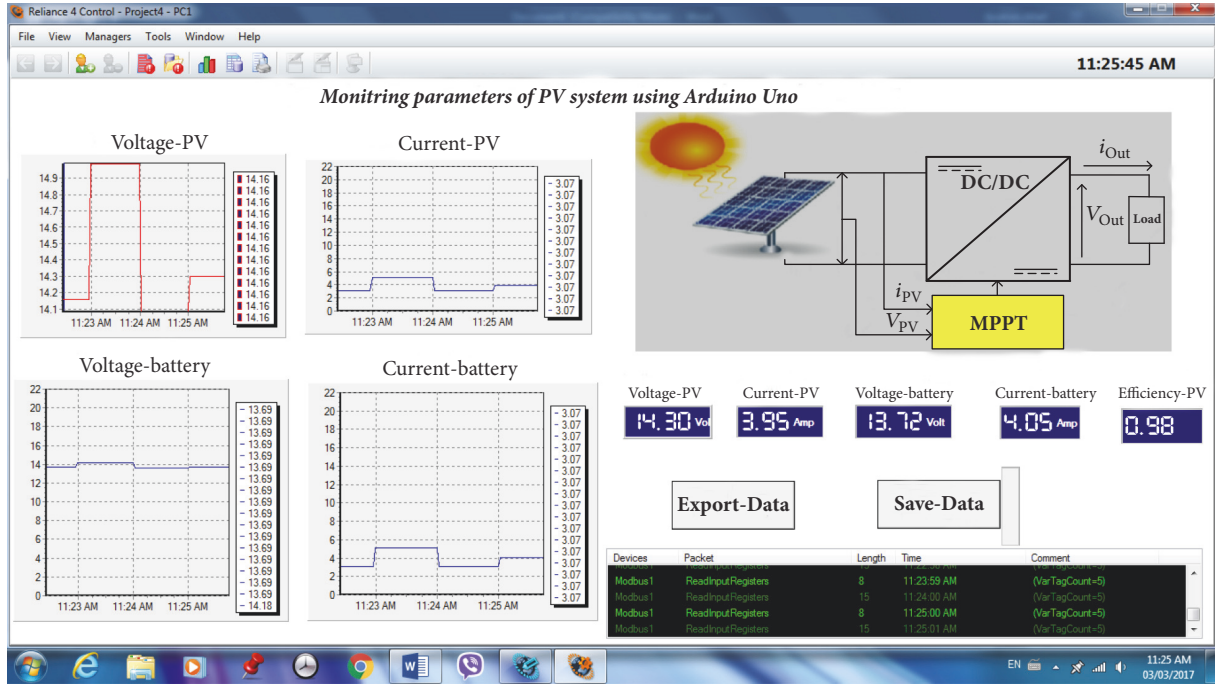


FIGURE 9: User interface of SCADA while running.

TABLE 3: Price components of SCADA system.

Number	Item	Quantity	Price
(1)	Current sensor	2	6.8
(2)	Voltage sensor	2	7.3
(3)	Arduino Uno	1	7.9
(4)	Software (IDE)	—	Free
(5)	Software (Reliance)	—	Free
(6)	Old small computer (PC) with Windows 7	1	60
Total amount			CA \$82

Table 3 shows the price (CA dollar) for whole components according to the amazon.ca website.

According to Table 3, we found that the whole price of SCADA system was CA\$82. This price seems cheap to design SCADA system for monitoring parameters of our system.

## 7. Results and Discussion

In this work, the proposed SCADA monitors a solar energy system and several experiments are carried out. The experiments cover the measurement error of the sensor systems which are installed to measure PV current and voltage, battery current and voltage, MMPT efficiency, and SCADA features.

The sensors that are used contain errors, so these errors are calculated with calibrated instruments, as listed in Table 4.

As can be seen in Table 4, the measurement error of current sensors was the highest. The error percentages of the PV current sensor and the battery current sensor are about

3.42% and 3.10%, respectively. Although the error percentages of both voltage sensors were quite low, they were closer to the calibrated instrument.

The monitoring tasks are displayed on the PC. They include the PV parameters as a graph and digital numbers and the MPPT efficiency as digital numbers. Figure 9 shows the user interface of SCADA after the system was operational.

The SCADA system is designed to make an update every minute. As shown in Figure 9, there are four figures: two of them observe the PV voltage and current and the other two monitor the battery voltage and current. The figure also shows that the SCADA system makes updates every minute.

The user interface of SCADA shows five icons displaying values of parameters as digital numbers, and they also make automatic updates every minute.

Our SCADA system has the feature of enabling all data to be easily saved on a computer as an Excel file. To save the data, the user just has to hit the Export-Data icon and then hit the Save-Data icon. These icons are programmed by script to save the data on a PC as an Excel file. Figure 10 shows a screenshot of data saved in Excel.

Also, user interface has a container that shows details. The Arduino connects with SCADA, and it gives warning if there is any error in connection.

The efficiency of MPPT was also monitored. It represents the output power of MPPT over the input power to MPPT. Figure 11 presents MPPT efficiency for various periods of time, with efficiency ranging between 1 and 0.8.

## 8. Conclusion

In this paper, a low-cost SCADA system was designed and built with Reliance SCADA software and Arduino Uno. The

TABLE 4: Measurement errors of sensor system.

Number	Sensor	Value	Calibrated instrument	Value	Measurement error
(1)	PV-voltage sensor	15.3 V	Voltmeter-ammeter	15.5 V	1.31%
(2)	Battery-voltage sensor	13.84	Voltmeter-ammeter	14.33	3.42%
(3)	PV-current sensor	3.22 A	Voltmeter-ammeter	2.77 A	1.62%
(4)	Battery-current sensor	3.66 A	Voltmeter-ammeter	3.55 A	3.10%

	A	B	C	D	E	F	G	H	I	J
1										
2		Voltage-PV:		Current-PV:		Voltage-B:		Current-B:		Efficiency:
3		17.13		11.66		15.38		12.01		0.92
4		17.13		11.66		15.38		12.01		0.92
5		17.13		11.66		15.38		12.01		0.92
6		16.89		11.27		15.18		11.42		0.91
7		16.89		11.27		15.18		11.42		0.91
8		16.89		11.27		15.18		11.42		0.91
9		16.89		11.27		15.18		11.42		0.91
10		17.13		11.66		15.38		12.01		0.92
11		17.13		11.66		15.38		12.01		0.92
12		17.13		11.66		15.38		12.01		0.92
13		16.99		11.27		15.35		11.71		0.93
14		16.99		11.27		15.35		11.71		0.93
15		16.99		11.27		15.35		11.71		0.93
16		16.99		11.27		15.35		11.71		0.93
17		16.99		11.27		15.35		11.71		0.93
18		16.99		11.27		15.35		11.71		0.93
19		16.99		11.27		15.35		11.71		0.93
20		16.99		11.27		15.35		11.71		0.93
21		16.77		10.64		15.13		10.79		0.91
22		16.77		10.64		15.13		10.79		0.91
23		16.77		10.64		15.13		10.79		0.91
24		16.77		10.64		15.13		10.79		0.91
25		16.77		10.64		15.13		10.79		0.91
26		17.4		12.15		15.33		11.91		0.86
27		17.4		12.15		15.33		11.91		0.86
28		17.4		12.15		15.33		11.91		0.86
29		17.4		12.15		15.33		11.91		0.86
30		17.4		12.15		15.33		11.91		0.86
31		17.4		12.15		15.33		11.91		0.86
32		17.4		12.15		15.33		11.91		0.86
33		17.4		12.15		15.33		11.91		0.86
34		16.77		10.64		15.18		10.79		0.91
35		16.77		10.64		15.18		10.79		0.91
36		16.77		10.64		15.18		10.79		0.91
37		16.77		10.64		15.18		10.79		0.91
38		16.77		10.64		15.18		10.79		0.91
39		16.77		10.64		15.18		10.79		0.91
40		17.13		11.66		15.38		12.01		0.92
41		17.13		11.66		15.38		12.01		0.92
42		17.13		11.66		15.38		12.01		0.92
43		17.13		11.66		15.38		12.01		0.92
44		17.13		11.66		15.38		12.01		0.92
45		17.13		11.66		15.38		12.01		0.92

FIGURE 10: Data saved in Excel file.

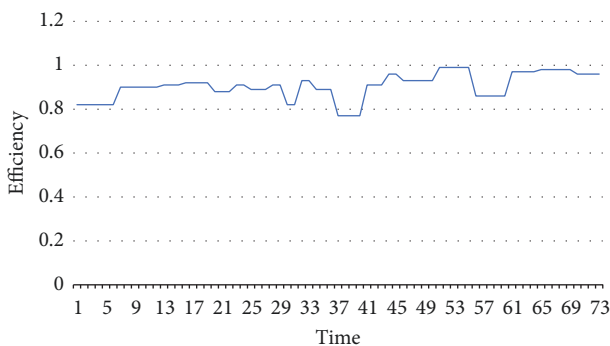


FIGURE 11: Efficiency of MPPT.

SCADA system was applied to a stand-alone photovoltaic system to monitor the current and voltage of PV and batteries. The results of the experiments demonstrate that SCADA works in real time and can be effectively used in monitoring a solar energy system. The developed system costs less than \$100 and can be modified easily for a different PV system.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would like to thank the Libyan government for providing financial support for this research.

## References

- [1] IRENA International Renewable Energy Agency, *Renewable Energy Technologies: Cost Analysis Series*, 2012.
- [2] C. Dumitru and A. Gligor, "SCADA Based Software for Renewable Energy Management System," *Procedia Economics and Finance*, vol. 3, pp. 262–267, 2012.
- [3] K. Kim, G. Parthasarathy, O. Uluyol, W. Foslien, S. Sheng, and P. Fleming, "Use of scada data for failure detection in wind turbines," in *Proceedings of the ASME 2011 5th International Conference on Energy Sustainability, ES 2011*, pp. 2071–2079, USA, August 2011.
- [4] L. Wang and K.-H. Liu, "Implementation of a web-based real-time monitoring and control system for a hybrid wind-PV-battery renewable energy system."
- [5] Figueiredo, João M. G., and M. G. Sá da Costa, "Intelligent sun-tracking system for efficiency maximization of International," in *Proceedings of the International Conference On Renewable Energy and Power Quality*, vol. 2008, Santander.
- [6] Figueiredo, J. M. J. Sa da Costa, and R. R. Ramalho, "Intelligent sun-tracking system for efficiency maximization of photovoltaic energy production," in *Proceedings of the International Conference On Renewable Energy and Power Quality*, vol. 1, Santander, 2008.
- [7] <http://www.sma.de/en/products/monitoring-control/sunny-view.html>.
- [8] <http://www.ebay.com.au/itm/SMA-Sunny-View-Colour-Touch-screen-Solar-Monitor-Screen-/251743364128>.
- [9] M. E. Treter, L. P. Pietta, P. Xavier, and L. Michels, "Data acquisition and cloud storage system applied photovoltaic systems," in *Proceedings of the 13th IEEE Brazilian Power Electronics Conference and 1st Southern Power Electronics Conference, COBEP/SPEC 2016*, Brazil, December 2015.
- [10] B. Ando, S. Baglio, A. Pistorio, G. M. Tina, and C. Ventura, "Sentinella: smart monitoring of photovoltaic systems at panel level," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 8, pp. 2188–2199, 2015.
- [11] P. A. V. E. L. Varacha et al., "Technical Devices for Supervising of a Household via Internet Based on Arduino Microcontroller," in *Proceedings of the 12th WSEAS International conference on Applied Informatics and Communications*, Istanbul, Turkey, 2012.
- [12] <https://www.reliance-scada.com/en/main>.



